Image Generation Design for Ground-based Network Training Environments

Brian Soderberg
Dr. Dale Miller
LORAL Advanced Distributed Simulation
Bellevue, Washington

Abstract

The SIMNET program represents the first large scale application of distributed simulation to tactical team training. A critical component in the virtual battlefield environment is the image generator (IG). After six years of operational use of an IG in this environment, the architectural design challenges for the next generation IG are seen to be quite different from those of other IG applications.

In this paper, the SIMNET program is briefly described. Its success has fostered widespread interest in distributed simulation technology and has provided a foundation for the requirements of the next generation of visual systems. The most challenging requirements for such systems specific to the needs of ground based tactical training environments are presented. An architectural approach to the design of an IG satisfying these requirements is then discussed. This includes a high level system diagram and the architectural concepts which provide solutions to high level occulting, localization of image complexity due to many moving models, dynamic allocation of processing resources, independent scalability of polygon and pixel processing, the need for standard interfaces, and simulation of special environmental effects such as tactical smoke. Architectural efficiencies and performance levels obtained with these methods are quantified.

I. Tactical Training Environments

SIMNET, an acronym for SIMulator NETworking, began in 1983 as a project focused on large-scale simulator networking by the Defense Advanced Research Projects Agency (DARPA). It was intended as a proof-of-principle technology demonstration of interactive networking for real-time, person-in-the-loop battle engagement simulation and wargaming. By 1990, its components consisted of about 260 ground vehicle and aircraft simulators, communications networks, command posts, and data-processing facilities distributed among eleven sites in the United States and Germany.

By intent, the SIMNET visual systems were to include only those aspects of the virtual visual environment which were deemed tactically significant. Second only to its major goal of providing sound tactical training based on distributed interactive simulation was the objective of providing this capability at sufficiently low cost to permit large scale deployment of the technology. At the time, visual systems for military aircraft simulators cost more than $1 million per visual channel; the goal, realized by the SIMNET project, was to reduce the cost by an order of magnitude. To meet the tactical training needs at this low cost, a visual system was needed which was high in visual complexity, with many moving models and special effects, but low in display complexity, with relatively few pixels, and a slow but constant update rate of 15 frames/sec.

The SIMNET project met its objectives, allowing interactive tactical team training and wargaming with thousands of military personnel, operating simulators of M1 Abrams tanks, M2/3 Bradley fighting vehicles, scout/attack helicopters, close air support fighter aircraft, and command and control functions. The success of this program, while significant in its own right, has proven that major advances in cost and performance of visual systems can be achieved by properly tailoring the IG architecture to the application. While SIMNET provided tactically meaningful training, additional realism in the virtual environment is expected to further enhance the training experience. Moreover, further advances in the price/performance ratio of visual systems are needed to continue the large scale deployment of the technology. While some of these required technological advances can be leveraged from developments in the microelectronics industry, key developments in algorithms and architectural approaches are needed to achieve these goals fully. In Section II, the requirements for the next generation IG for tactical training are presented. Section III then gives architectural approaches to satisfy these requirements. Table 1 is a matrix showing the correlation of approaches to requirements. Finally, in section IV, the benefits of these methods are quantified.

0-7803-1363-1/93 $3.00 © 1993 IEEE

Requirements / Approaches	Quad level occulting	Tri-level fixed interleave	Distribution buses	Partial/full antialiasing	Open architecture	Volumetric smoke
Accomo-date ground vantage points with many moving models	X	X	X	X		
Independent of spatial distribution of scene complexity	X	X	X			
Static & dynamic allocation of processing resources		X	X			
Support high ratios of polygon to pixel processing			X			
Open, standard interfaces and buses					X	
Support special environ-mental conditions						X

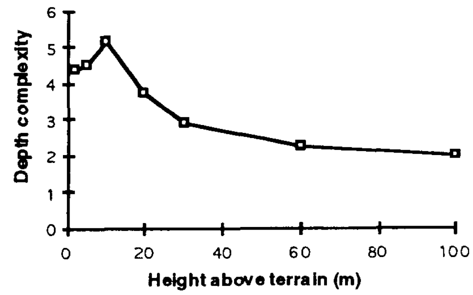
Table 1 Correlation of IG architectural approaches to visual system requirements

II. Visual requirements of tactical training environments

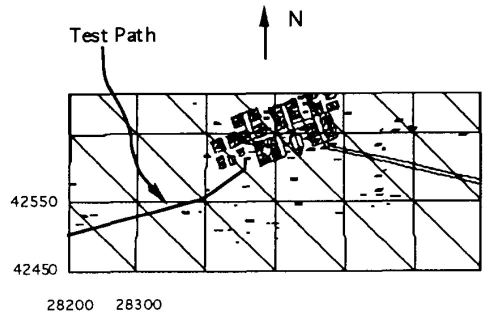
Requirements for visual systems which satisfy the needs of tactical training using distributed simulation differ in several fundamental ways from the requirements imposed by other training applications. In this section, six requirements areas are described which pose challenging design problems whose solutions are critical for effective distributed interactive tactical team training. These areas are presented in terms of the requirement statement and a trade study of design implications or options.

Requirement 1: The visual system shall suffer no image or frame rate degradation as a function of the height of the vantage point above the terrain surface or the number of moving models present in the scene.

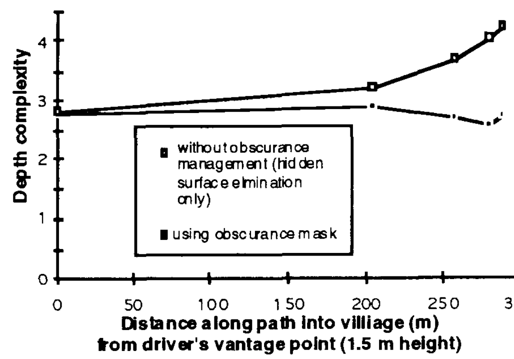
Tradeoff Analysis. In regions containing rugged terrain or many cultural features, depth complexity increases significantly as the height of the eye point approaches the terrain surface as shown in Figure II-1. The data shown were acquired near the village of Bullenheim in the German AGPT database. This is the fundamental reason that IGs designed for the flight simulation market suffer pixel overload in ground applications.



(a) Depth complexity increases significantly for near ground viewpoints. This data was collected above a ground point near Bullenheim in the AGPT database.



(b) Plan view of Bullenheim Village from 1300m, showing the path into the village along which the depth complexities in (c) are calculated.



(c) Obscurance management provides leveling of required computational depth complexity independent of scene complexity.

Figure II-1 Data from the German database shows (a) depth complexity requirements are greatest near ground, and (b) obscurance management reduces depth complexity in complex scenes.

Visual systems that are optimized for air vehicle simulators solve polygon occulting off-line for fixed objects like land, natural features, and cultural features. One frequently used polygon sorting technique is generally known as the binary separation plane method. If all objects are fixed, this technique is excellent for complex,

ground based scenes. However, when faced with moving models, such approaches must then augment the off-line occulting with a pixel-level occulting process such as Z- or R- buffering. Because such an architecture is not optimized for extensive real-time occulting, frame rate reduction can be used to successfully resolve depth complexity when faced with many moving models. A slower and irregular frame rate has potential for a negative training impact, especially for gunnery tasks. However, negative training for the gunner is likely to happen for such systems because gunnery is a task during which a large number of moving targets can be expected to predominate in the scene.

Figure II-2 shows the expected degradation in frame rate for an IG with mixed occulting consisting of off-line binary separation plane and real-time R-buffer techniques. The example represents an IG that is optimized to process scenes with up to 30% polygon content consisting of moving models. Note the rapid fall-off of frame rate as the content increases beyond 30%. If the IG is to be modified to handle a much larger percentage of moving models, the pixel processor and bus structure must be dramatically upgraded. As such, the IG may as well be designed for 100% real-time occulting from the start.

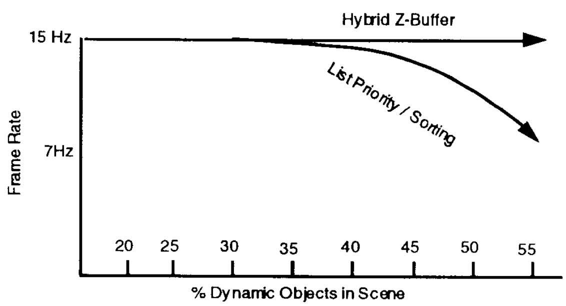


Figure II-2 The hybrid z-buffer architecture provides constant performance compared with list priority and sorting architectures.

The architecture described in the section III uses real-time quad-level occulting to handle occulting of all objects, whether fixed or moving. Quad-level means that hidden area modules (large groups of objects in a geographic region), single objects, polygons, and pixels are sequentially filtered-out in real-time. The mask buffer operates in the Polygon Processor to provide the first three levels of occulting. Then the hybrid Z-buffer operates in the Pixel Processor to provide the final pixel-level occulting.

Requirement 2: The visual system shall optimally utilize all computational resources in rendering a view *independently* of the spatial distribution of the scene complexity.

Tradeoff Analysis. One of the primary architectural goals for the proposed approach has been to distribute evenly the computational load among the processing resources. Pipeline approaches have been constrained to associate processing resources with display regions or channels. When image complexity becomes localized, some computational resources become overloaded while others are idle. Given the unpredictable nature of scene content in a distributed interactive simulation, scene complexity is frequently restricted to a localized viewport region or channel. A pipeline architecture will result in visual anomalies which are intrusive to the training process.

The architecture described in section III utilizes a Tri-Level Fixed Interleave paradigm. This provides medium grain parallelism for polygon, tiling, and pixel operations. Input data at each stage are divided into spatially distributed subsets that are interleaved among the parallel processors using a fixed, precalculated mapping which minimizes the correlation of local scene complexity with any one processor. Thus the loading is leveled implicitly, without active real-time intervention.

Requirement 3: The visual system shall be configurable to match the specific polygon and pixel processing requirements for the application; and shall allocate processing resources in real time to support dynamic variation in scene complexity from channel to channel.

Tradeoff Analysis. To meet these goals, the visual system should be reconfigurable through modularity, scalability, and real-time allocation of processing resources across channels. If a given application needs more polygon capacity, less pixel capacity, and more channel capacity than a nominal configuration, such needs should be achieved by independently adjusting the polygon, pixel, and video generation resources accordingly. With a fixed pipeline architecture in which the output of each stage feeds directly into the next, tuning the ratios among polygon, pixel, and display capacities is impossible.

In a dynamic battlefield environment, static allocation of processing resources on a per

channel basis will yield overload on some channels while resources associated with other channels are underutilized. Traditionally, architectures have been designed so that computational resources are statically assigned to display channels. Such an approach will work well for the 3-channel configuration shown in Figure II-3(a). Here, the polygon loading due to the moving vehicles is uniform across the three channels, and each channel is at its maximum polygon capacity.

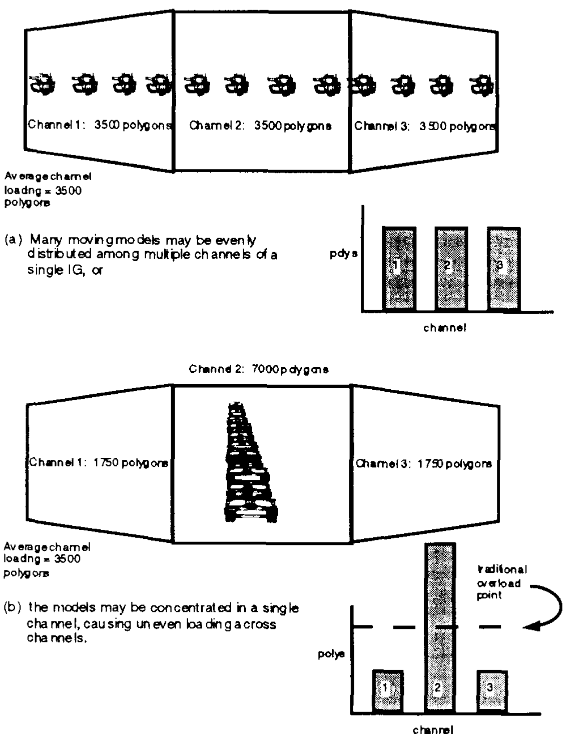


Figure II-3 A dynamic battlefield environment will cause uneven distribution of polygons to channels. Dynamic reallocation of processing resources between channels will address this problem.

Given the unpredictable motion of vehicles in a dynamic battlefield environment, such uniform loading of polygons among channels is unlikely. The same scene from a different vantage point is shown in Figure II-3(b). Here, all the polygons from the moving models are concentrated in the center channel.

The proposed architecture utilizes a shared memory multiprocessor approach. During a computation frame, as a processor completes the processing of its assigned database region, it will check the loading of its neighboring processors which are assigned to other channels. It then offloads processing from other busy processors.

Thus the full system processing capabilities can be utilized, meeting any situation in which some channels are overloaded but the *average* channel loading is within the system capacity.

Requirement 4: The visual system shall be configurable to support a ratio of polygon performance relative to pixel performance of at least six polygons per thousand pixels. This will preserve the three-dimensional complexity of the ground and near-ground scenes while providing high resolution.

Figure II-4 shows the historical polygon and pixel performance requirements growth of the world's only distributed interactive tactical team trainer programs to date. Note that while the polygon and pixel performance requirements have grown, the ratio of polygon to pixel performance has remained within a narrow region of about 2 to 6 polygons per thousand pixels. Requirements for the next generation of distributed tactical team trainers have remained within these bounds.

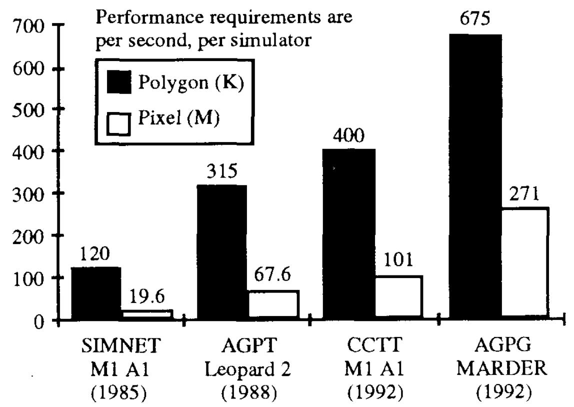


Figure II-4 Ratio of polygon to pixel performance has remained relatively constant for all distributed simulation training programs. This ratio is about four times higher than the ratio used for aviation simulation.

Visual systems optimized for the air vehicle market have not been balanced for this required range of polygon to pixel performance ratio, typically providing a relatively large number of pixels compared to polygons. The ratios lie within a range of about 0.4 to 1.5 polygons per thousand pixels. This low ratio of polygons to pixels is correct for air vehicle simulators because three-dimensional complexity and depth complexity are very low when the real-world is simulated for higher altitudes.

Because an image generator's board and bus architecture determine its optimal operating range

of polygons to pixels, for a machine to be optimized for distributed interactive tactical team training it must be designed for this mission from the start. In addition, such a system should be modular and scalable to allow each IG rack to be configured for its module's precise requirements.

Requirement 5: The visual system shall utilize open, standard interfaces and buses.

Tradeoff Analysis. Image generators have been developed as special purpose high performance computers. Because of the complex, custom nature of these machines, interfaces have been largely proprietary, and custom buses have often been required to meet performance demands.

As distributed training simulation becomes more widespread, industry standards are necessary for interoperability and incremental component improvements. Standards are required for databases, communication protocols, IG and network programming interfaces, buses, and multiple sourced COTS boards. High performance industry standard buses now exist which can meet performance requirements. A uniform, standard application programming interface from the simulation host to the image generator should be defined and adopted which will (1) allow the simulation host to manipulate easily the higher level objects of the virtual environment, and (2) prevent the simulation host code from becoming tightly bound to the particular image generator model.

Requirement 6: The visual system shall provide direct support for all environmental conditions of tactical significance, including local obscurants (e.g., tactical smoke, dust clouds), local illumination sources (e.g., flares, headlights), continuous time of day, sensor simulation and various weather conditions.

Tradeoff Analysis: These environmental conditions are of critical tactical importance during actual warfare, and should be faithfully represented in the next generation of tactical team trainers. Neither computer hardware nor computer graphics algorithms have permitted such faithful representation heretofore. Advances in both areas now permit such simulation capabilities. Of particular tactical importance and technical difficulty is the simulation of three dimensional volumetric items for the representation of local obscurants. To provide an adequate training experience, such a representation must provide proper representation of a volumetric item from any vantage point (either internal or external to the obscuring volume) and must properly obscure all other scene elements (in front of, inside, or behind the obscuring volume).

III. Image Generation Architectural Approaches

Six architectural concepts were developed to address the visual requirements described in section II. They include: (1) Quad-level Occulting, (2) Tri-level fixed interleaving, (3) Distribution Buses for processing, (4) Hybrid Z-buffer with partial/full anti-aliasing, (5) Open Architecture / Standards, and (6) Volumetric Smoke Processing.

Following the trade studies and analysis described in section II; key architectures and system strategies were developed specifically to address the task of visualizing complex battlefield environments in real-time. In this section each of the critical design concepts which influenced the visual system architecture will be described.

Concept 1 - Quad Level Occulting. This technique rejects regions, objects and polygons early in the graphics processing pipeline in order to reduce to number of required pixel processing compute elements in a visual system.

In networked training systems, the battlefield environment is one of high complexity with exceptionally high pixel processing requirements due to looking through layers of vegetation, smoke, haze, buildings and mountains from a ground vehicle. The excessive pixel processing due to the layers of rendered objects requires a large number of pixel processors, substantially increasing the required pixel processing components.

We observed during distributed simulation testing that an IG can experience excessive pixel load conditions. However, we also assessed that during high loads as much as 75% of the pixels were actually obscured from view by nearer objects (see figure II-1). This simple observation drove us to investigate methods for rejecting obscured objects as early as possible in the computational process.

The technique chosen is a simple method which rejects hidden objects early in the graphics processing pipeline. By rendering objects generally from front to back, we can detect when entities are completely hidden by nearer objects.

For maximum efficiency, this high level occulting is performed at three levels; regions, objects and polygons. The final, pixel-level occulting is performed with the hybrid Z-buffer. This sequence accounts for all four levels of occulting, or Quad Level Occulting. (Figure III.b).

High Level Occulting

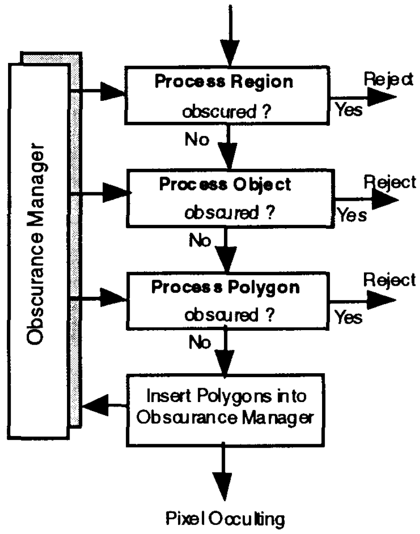


Figure III.b High level occulting performed in the polygon processor rejects regions, objects and polygons prior to costly pixel processing.

Concept 2 - Tri-level Fixed Interleave Processing levels the loading across all database, polygon, and pixel processors.

In networked training systems, the visual system processing requirements are such that millions of polygons and hundreds of millions of pixels must be processed every second. This enormous processing load requires many resources working in parallel on polygon and pixel calculations. The challenge in this graphics environment is to develop a technique that levels the loading of many parallel processing resources. Without load leveling, each resource must be designed to handle the worst case, driving system costs excessively high.

The Fixed Interleave Processing architecture divides the processing job into a pseudo-random, fixed interleaved pattern of regions which are assigned to the different processors. Each processor will process many of these randomly located regions. The assignment of processors to regions is a fixed repeating pattern as shown in Figure III.c.

Level 1 - High Level Fixed Interleave Database Processing: The highest level of fixed interleave processing is the allocation of fixed-size database regions (area modules) to polygon processors. This solves two problems; The first being the parsing out of polygon processing to parallel engines. The second is allowing a specific database region to be processed in a single processor, even when required across multiple channels. This alleviates the need for additional memory and processing resources when a region must be processed for multiple channels.

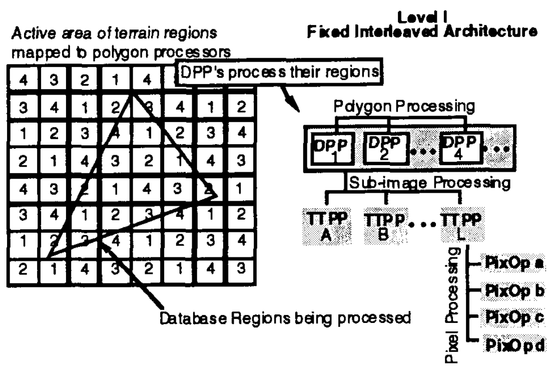


Figure III.c Interleaved database regions are distributed to Polygon Processors for implicit Load Leveling.

Level 2 - Image Sub-region Fixed Interleave Processing: At this level, the displayed image is divided into small sub-regions that are assigned to the tilers in a pseudo-random, but fixed manner. This levels the load across all pixel processors. Typically, tilers process a large contiguous area of the image. This technique suffers overload when all the processing falls into one pixel processors region of the image. Our technique solves this problem by having small sub-regions (commonly 64x64 pixels) and assigning many sub-regions from different channels to a single Tiler. Each tiler maintains equal loading even with localized regions of high pixel processing (Figure II.d).

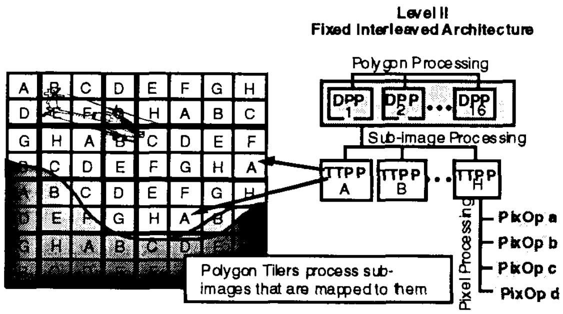


Figure III.d Fixed Interleave Image Sub-region mapping to tilers ensures implicit pixel load leveling.

Level 3 - Two by Two Pixel, Fixed Interleave Processing: The third level of fixed interleaving is at the lowest pixel processing level in the system. The image is further divided into 2x2 pixel blocks spread across multiple pixel operators on a tiler. This fine grain parallelism, in a fixed pseudo-random orientation, ensures equal loading across all pixel processing resources (See figure III.e).

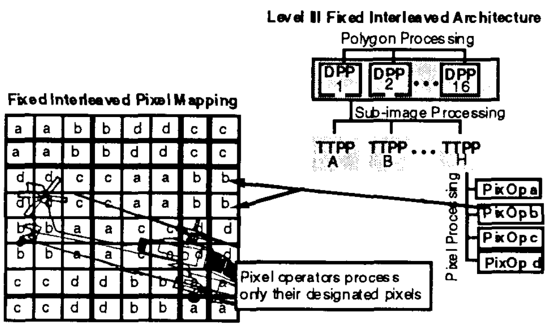


Figure III.e Fixed Interleaved 2x2 Pixel mapping to pixel operators ensures implicit load leveling.

This tri-level fixed interleaved architecture maximizes equal loading across all processing resources under all conditions, providing consistent maximum system performance required for the multi-channel training in complex battlefield environments.

Concept 3 - Polygon and Pixel Distribution Buses: Maximum IG configurability, expansion, and efficient processing is required for the variety of simulator configurations in networked training. To accomplish this we implemented distribution buses between all graphics processing stages.

A variety of polygon and pixel processing capacities are necessary for the channels of each simulation module. In some cases high pixel processing capacities are required for high resolution views. In other cases high polygon processing capacities are required for wide fields of view. In some cases, polygon processing demand changes dynamically from frame to frame across several channels. These variations in processing allocation and dynamic processing requirements must be addressed in the system design.

Most graphics systems have rigid pipelined architectures. The database engine feeds the polygon processor, which feeds the pixel processor and finally the video processor. The entire graphics pipeline is then duplicated to provide the required processing levels. The major drawback in this architecture is that it does not provide configurability or expansion in a single processing area. For example, if an application requires triple the polygon processing of a single pipeline, but does not require additional pixel processing the entire graphics pipeline cost must be tripled.

To eliminate the inefficiencies associated with configuring traditional pipelined architectures, and to allow dynamic process reallocation, we implemented *Distribution Buses*. This bus architecture allows all database, polygon, pixel, and video processors to feed among themselves, many to many. This modularity provides optimum matching of polygon and pixel requirements among the diverse module, console, and battle scenario requirements. Moreover, the polygon bus prevents excessive poly processing demands from overloading a local set of processors.

Making use of the Fixed Interleave Processing and broadcasting on the polygon distribution bus, few or many pixel processors can be configured as required for a particular application. Crossbar interconnects are used across all polygon processors to provide additional load leveling. The crossbar allows for poly process resource allocation across all processing nodes. In the case that one node is overloaded and another is underutilized, the extra processing required is provided by the underutilized node. This technique is used to level processing demands across all channels. This is very useful in distributed simulation, where if one channel is heavily loaded, others tend to be underutilized. The polygon process allocation completely addresses this irregular loading case. See the polygon and pixel distribution bus architecture in figure III.f.

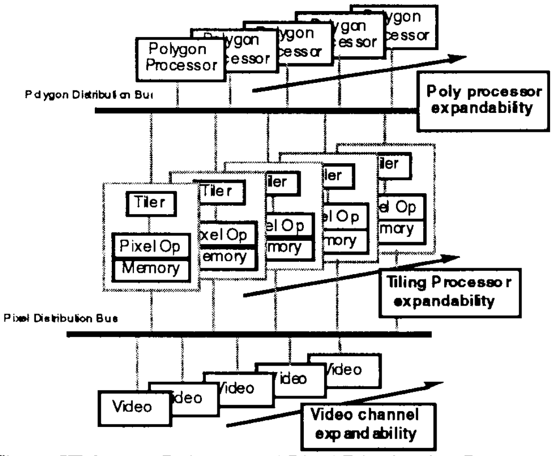


Figure III.f Polygon and Pixel Distribution Buses allow for modular expansion of required resource polygons, pixels or channels.

With use of the Fixed Interleave Pixel Processing and the Pixel Distribution Bus, one to many video channels can be fed from one to many Pixel Processors. As with the other system components, the video channels can be optimized for the particular application without incurring additional system overhead.

Concept 4 - Hybrid Z-buffer with Partial / Full Buffer Anti-aliasing: We developed a Hybrid Z-buffer technique with Partial / Full Anti-aliasing (patent pending) to address the hidden surface elimination requirements of complex dynamic battlefield environments.

In the ground-based armor simulation environment many dynamic effects and actions take place, such as high numbers of moving vehicles through detailed terrain models. Moreover, dynamic ballistic effects, drifting smoke, and dust clouds, to name a few, have important tactical significance and must be visualized. All of these effects comprise a complex set of opaque, transparent and varying density volumetric graphics primitives to render.

Traditional Z or range buffer occulting methods can only be as good as the resolution of the stored Z values. This can be a problem when objects get closer to each other than the Z resolution can handle. The occultation then produces artifacts. Another area of failure is handling coplanar polygons such as a road on terrain. Typical z buffering must accomplish this in one of two ways; Either break up the terrain polygon to insert the road incurring substantial extra polygon processing or to inappropriately float the road above the terrain to alleviate the Z resolution problem. Neither of these solutions are optimal.

Our technique combines Z-buffering with a priority system to solve the accuracy and coplanar problems. This method is based on a simple concept of prioritizing objects such that when the distance between two objects is small, the higher priority object should be viewed in front.

Anti-aliasing Z-buffers is traditionally a difficult task to accomplish without utilizing expensive supersampling solutions of tiling 8 or 16 sub-pixels per pixel. This supersampling approach increases the pixel processing requirements considerably. We developed a new technique which operates separately on partial pixels (either partially covered or partially transparent) and fully covered opaque pixels.

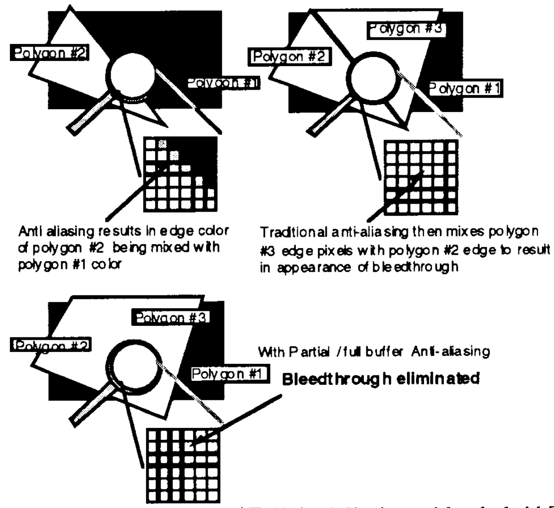


Figure III.g Partial / Full Antialiasing with a hybrid Z-buffer minimizes the image quality anomalies ("bleed-through") associated with traditional Z-buffer antialiasing techniques.

This technique nearly eliminates all the typical artifacts associated with anti-aliasing with a Z-buffer without using the expensive supersampling techniques. The image quality exceeds that provided by 16:1 supersampling.

Concept 5 - Open Architecture/Visual System Standards: We chose to implement industry standards in interfaces, buses and networks wherever possible. This allows the distributed system users to work from a common base of knowledge and resources. Additionally, future inclusion of new simulation models and applications will be simplified. Without Standards compliance, life cycle of simulation

applications, databases, and hardware will be much reduced.

CIG Logical Interface Package (CLIP) - We proposed a graphics interface definition for the Distributed Interactive Simulation (DIS) IG interface. Using this uniform extensible interface, makes use of a high-level command set which directly addresses the needs of DIS. Various simulation modules can be easily interfaced to the IG through this package.

Internal to the IG, standard high performance buses are utilized. The real-time load bus makes use of VME64 for standard interaction of most processing resources for control of real-time parameter driven functions such as haze functions, gamma or laser range return. The polygon distribution bus makes use of a Futurebus+. Network interfaces use standard Ethernet.

Concept 6 - Volumetric Smoke Processing - To properly manage localized overlapping volumetric smoke processing within a hybrid Z-buffer architecture, we came up with a technique which manages multiple smoke volume buffers in addition to the standard color and depth frame buffer elements. The smoke volume buffers are utilized to allow proper attenuation of non-smoke volume objects which fall within and beyond volumetric entities. This is done through the storage of multiple volumetric item density and range values with the frame buffer data to allow for applying varying levels of transmittance through smoke models. The complex geometry of the smoke model is stored within the texture map, allowing complex smoke shapes to be modeled with very few polygons. For example, with the volumetric based rendering, a large billowing smoke plume can be modeled realistically with dozens of polygons rather than hundreds with traditional polygon rendering.

IV. Architectural and Performance Results

This section documents the architectural and performance benefits of the IG architectural concepts described in sections II and III.

High-level Occulting - As described in section II and III, this method was developed to help alleviate the high depth complexity associated with complex environments viewed from ground and near-ground vantage points. We simulated the CIG architecture and got promising results.

The goal of this method is to cap the pixel processing level over a range of image complexities. When depth complexity is low, such as 2 to 3, this method doesn't substantially reduce the required pixel processing. But as the depth complexity climbs to levels such as 4 to 6, the technique contains the computational depth complexity to less than 3. Figure IV.a shows the resulting depth complexity with respect to elevation of the vantage point. It demonstrates that the required pixel processing gets excessively high as the vantage point approaches the terrain level. With the high-level occulting, the computed depth complexity is contained to less than 3, becoming effective as the actual depth complexity increases. This is exactly the desired effect needed to allow IG's to be configured to render visually complex scenes with minimal pixel processing hardware.

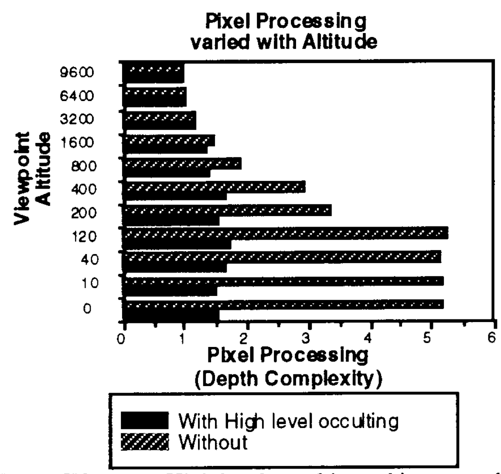


Figure IV.a High level occulting achieves nearly a 3X decrease in pixel processing associated with ground based vantage points for this case.

Figure IV.b demonstrates another case study, showing the depth complexity with respect to heading. It shows the depth complexity is low for many viewing directions. At headings looking towards a village, the depth complexity becomes excessively high. Once again, in the cases where the depth complexity is highest, the high-level occulting contains the pixel processing to a manageable level. The elegance of this technique is that the higher the complexity of the environment, the better the method works.

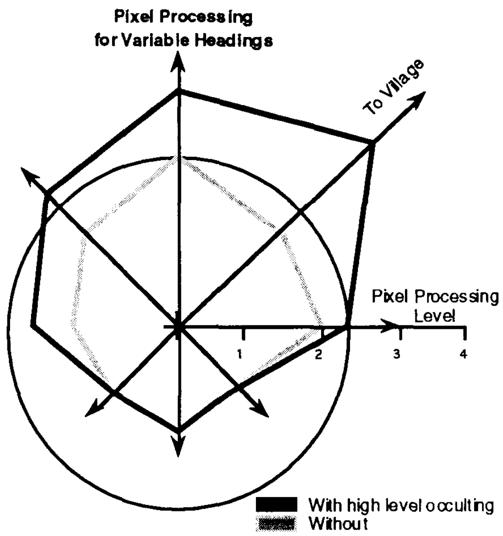


Figure IV.b High pixel processing associated with irregular loading is minimized with the high level occulting technique.

Fixed Interleaving / Distribution Buses - As described in section III, these methods were developed to allow for maximum utilization of all processing resources of the IG. With this architecture all processors stay busy, even when the processing load is highly irregular.

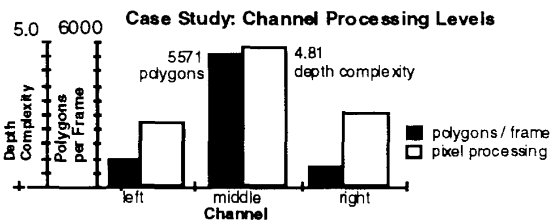


Figure IV.c This polygon and pixel processing loading distribution is representative of a 3 channel IG configuration case study.

Figure IV.c is a case study showing the required polygon and pixel levels for 3 viewing channels of a battlefield environment. The center channel looks at a dense village and bridge scene, with much higher processing requirements than the two adjacent channels. With the fixed interleaving and distribution buses, the IG naturally processes the aggregate processing level of all channels with equal load on all processing elements. A traditional, fixed pipeline architecture must configure for the worst case channel. The fixed interleaved architecture is compared with the traditional fixed graphics pipeline in figure IV.d.

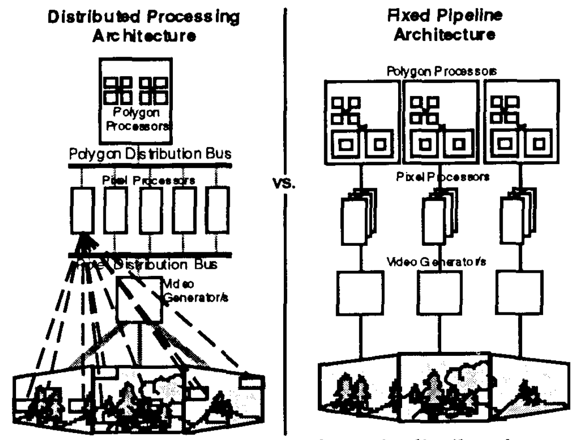


Figure IV.d This diagram shows the distributed processing architecture requires far fewer boards for the case study of figure IV.c.

In this case study, the traditional fixed pipeline architecture requires twice the number of processing elements to handle the potential worse case condition in any channel. In our fixed interleaved architecture with distribution buses, we only require processing for the aggregate loading of the three channels. Figure IV.e details the number of boards required with the two architectures, to handle the case study shown in figure IV.c.

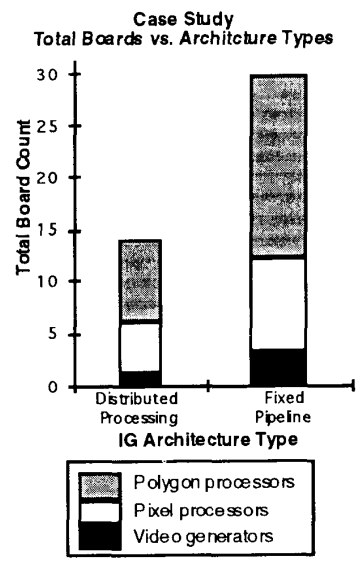


Figure IV.e The distributed processing architecture requires only 14 boards compared to 30 for the fixed pipeline architecture. These numbers correlate to the case study in figure IV.c.

The second benefit of the distributed processing architecture is the implicit modularity. This

allows the IG to be effectively configured for all applications independent of polygon to pixel to channel processing requirements. As shown in figure 4.d, the IG can independently add polygon, pixel and channel resources exactly matching application requirements for a specified IG application. Fixed pipeline architectures force a specific polygon / pixel processing ratio, and expansion in any area must accept the associated overhead of the other boards in the pipeline.

Hybrid Z-buffer / Volumetric Smoke Processing - Our hybrid Z-buffer occulting and anti-aliasing solves two problems. It allows for processing objects which require occulting at accuracy's higher than the resolution of the depth values and the handling of coplanar polygons.

The second area of benefit is associated with the integrated partial / full buffer anti-aliasing. It requires much less processing and memory than the traditional supersampling techniques used for anti-aliasing a z-buffer. Figure IV.f shows the comparison of the partial / full buffer anti-aliasing with 8 and 16 supersampling systems.

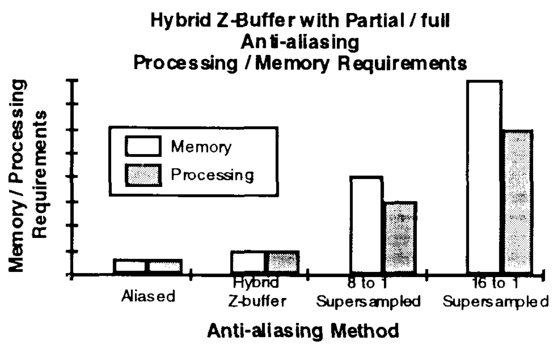


Figure IV.f The partial / full buffer anti-aliasing method takes much less memory and processing when compared to traditional supersampling techniques.

Volumetric smoke processing provides several benefits. The greatest result is the added feature of true volumetric smoke, dust, and clouds which interact properly with other battlefield elements. This method is far superior to the traditional polygonal modeling methods which attempt to model volumetric effects. First, our technique models smoke with very few polygons, encoding the 3D shape in the texture map. Figure IV.g shows the processing requirements to model smoke with traditional polygonal modeling vs. our volumetric texturing methods. A newer method known as voxel (volumetric element) processing has been explored for rendering local obscurant objects. The major disadvantage to this technique is the enormous level of memory and pixel processing required to render the 3D image data. The resulting rendered smoke imagery using voxels also tends to be blocky in nature, limited by the 3D image storage. For these reasons we rejected this approach.

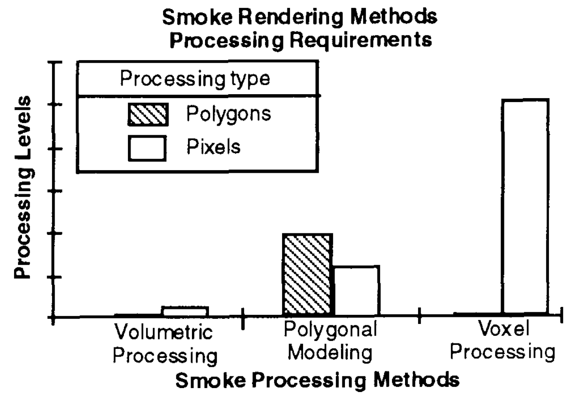


Figure IV.g The volumetric processing technique reduces required processing and is more realistic compared to other approaches.

Even with the substantially higher processing done with polygonal modeling, it still falls far short of modeling realistic volumetric effects.

Authors

Brian Soderberg is a LORAL Division Scientist and the manager of the CIG development and algorithm research group. Mr. Soderberg has ten years experience in computer graphics and image generator research and development. His research focuses on development of new algorithms for image generation to support LORAL's ongoing CIG development programs.

Dr. Dale D. Miller is the Program Engineer for advanced CIG development at LORAL and the manager of Software Engineering in the Visual Systems Engineering department. Dr. Miller holds a Ph.D. in mathematics and has twelve years experience in graphics, image processing and computer vision.

References

1. R. Bess and B. Soderberg, "Battlefield smoke: a new dimension in networked simulation," in *Proc. 13th Interservice / Industry Training Systems Conf.*, Orlando, 1991.

2. Dr. Duncan Miller, "The SIMNET Architecture for Distributed Interactive Simulation", in *Summer Simulation Conference*, Baltimore, Maryland, July, 1991.

3. R. Bess, "Image generation implications for networked tactical training systems," in *Proc of the 1992 Image VI Conf.*, Scottsdale, 1992, pp. 77-88.

4. Dale Miller, "CLIP - a proposed uniform standard image generator interface for DIS," in *Seventh Workshop in Standards for Interoperability of Defense Simulations*, Orlando, 1992.